

CLAIMS

What is claimed is:

1 1. A portable thread environment comprising:
2 an application programming interface configured to support multiple
3 application program tasks, wherein each task is either a preemptive task
4 comprised of preemptive threads or a cooperative task comprised of cooperative
5 threads;
6 host adaptation logic for communicatively interfacing said cooperative
7 tasks, preemptive tasks, cooperative threads and preemptive threads with a host
8 processing environment; and
9 a scheduler configured to determine an execution order of cooperative
10 threads and preemptive threads based on each cooperative thread's and
11 preemptive thread's priority levels.

1 2. The portable thread environment as in claim 1 wherein threads of the
2 same preemptive task or cooperative task have different priority levels.

1 3. The portable thread environment as in claim 1 wherein a currently
2 running preemptive thread is suspended by a higher priority thread.

1 4. The portable thread environment as in claim 1 wherein a preemptive
2 thread may suspend a currently running lower priority thread.

1

1

1 5. The portable thread environment as in claim 1 wherein a currently
2 running cooperative thread is suspended when a preemptive higher priority
3 thread is requested.

1 6. The portable thread environment as in claim 1 wherein a currently
2 running cooperative thread of a task is not suspended when a requested
3 cooperative thread of the task is requested.

1 7. The portable thread environment as in claim 1 wherein a currently
2 running cooperative thread of a first task is suspended when a requested
3 cooperative thread of a second task and having a higher priority level is
4 requested.

1 8. The portable thread environment as in claim 1 wherein a first thread
2 may change a priority level of a second thread.

1 9. A method for porting an application from a first host environment to
2 a second host environment, comprising:
3 modeling functions supported by said application-specific hardware in
4 the form of a first set of tasks within said application;
5 removing said first set of tasks from said application;
6 loading said application without said first set tasks to said second host
7 environment, said application-specific hardware providing said functions
8 provided by said first set of tasks in said first host environment; and
9 configuring said first set of tasks and second set of tasks to communicate
10 by passing a set of messages in said first host environment, wherein one or more
11 of said set of messages are also used to provide communication between said

12 first set of tasks and said application-specific hardware in said second host
13 environment.

1 10. The method as in claim 9 further comprising:
2 interrupting the first set of tasks.

1 11. The method as in claim 10, wherein interrupting the first set of tasks
2 further comprises:
3 determining an execution order of cooperative threads and preemptive
4 threads based on each cooperative thread's and preemptive thread's priority
5 levels.

1 12. The method as in claim 11, further comprising:
2 sustaining a currently running preemptive thread when a cooperative
3 thread or preemptive thread of lower or equal priority is requested.

1 13. The method as in claim 11, further comprising:
2 suspending a currently running cooperative thread when a preemptive
3 thread is requested.

1 14. The method as in claim 11, further comprising:
2 sustaining a currently running cooperative thread of a task when a
3 requested cooperative thread of the task is requested.

1 15. The method as in claim 11, further comprising:

2 suspending a currently running cooperative thread of a first task when a
3 requested cooperative thread of a second task and having a higher priority level
4 is requested.

1 16. A portable application environment, comprising:
2 means for modeling functions supported by said application-specific
3 hardware in the form of a first set of tasks within said application;
4 means for removing said first set of tasks from said application;
5 means for loading said application without said first set tasks to said
6 second host environment, said application-specific hardware providing said
7 functions provided by said first set of tasks in said first host environment;
8 means for configuring said first set of tasks and second set of tasks to
9 communicate by passing a set of messages in said first host environment,
10 wherein one or more of said set of messages are also used to provide
11 communication between said first set of tasks and said application-specific
12 hardware in said second host environment; and
13 means for passing the set of messages from a first thread to a second
14 thread.

1 17. The portable application environment as in claim 17 further
2 comprising:
3 means for interrupting the first set of tasks.

1 18. The portable application environment as in claim 17 further
2 comprising:

1 23. The portable application environment as in claim 17 further
2 comprising:
3 means for suspending a currently running cooperative thread of a first
4 task when a requested cooperative thread of a second task having a higher
5 priority level is requested.

1 24. A computer-readable medium having stored thereon a plurality of
2 instructions, said plurality of instructions when executed by a computer, cause
3 said computer to perform:

4 modeling functions supported by said application-specific hardware in
5 the form of a first set of tasks within said application;

6 removing said first set of tasks from said application;

7 loading said application without said first set tasks to said second host
8 environment, said application-specific hardware providing said functions
9 provided by said first set of tasks in said first host environment;

10 configuring said first set of tasks and second set of tasks to communicate
11 by passing a set of messages in said first host environment, wherein one or more
12 of said set of messages are also used to provide communication between said
13 first set of tasks and said application-specific hardware in said second host
14 environment; and

15 passing the set of messages from a first thread to a second thread.

1 25. The computer-readable medium of claim 24 having stored thereon
2 additional instructions, said additional instructions when executed by a
3 computer, cause said computer to further perform:

4 interrupting the first set of tasks.

1

1 26. The computer-readable medium of claim 25 having stored thereon
2 additional instructions, said additional instructions when executed by a
3 computer, cause said computer to further perform:

4 determining an execution order of cooperative threads and preemptive
5 threads based on each cooperative thread's and preemptive thread's priority
6 levels.

1 27. The computer-readable medium of claim 25 having stored thereon
2 additional instructions, said additional instructions when executed by a
3 computer, cause said computer to further perform:

4 sustaining a currently running preemptive thread when a cooperative
5 thread or preemptive thread having lower or equal priority is requested.

1 28. The computer-readable medium of claim 25 having stored thereon
2 additional instructions, said additional instructions when executed by a
3 computer, cause said computer to further perform:

4 suspending a currently running preemptive thread when a higher priority
5 cooperative thread or higher priority preemptive thread is requested.

1 29. The computer-readable medium of claim 25 having stored thereon
2 additional instructions, said additional instructions when executed by a
3 computer, cause said computer to further perform:

4 suspending a currently running cooperative thread when a higher priority
5 preemptive thread is requested.

1 30. The computer-readable medium of claim 25 having stored thereon
2 additional instructions, said additional instructions when executed by a
3 computer, cause said computer to further perform:
4 sustaining a currently running cooperative thread of a task when a
5 requested cooperative thread of the task is requested.

1 31. The computer-readable medium of claim 25 having stored thereon
2 additional instructions, said additional instructions when executed by a
3 computer, cause said computer to further perform:
4 suspending a currently running cooperative thread of a first task when a
5 requested cooperative thread of a second task and having a higher priority level
6 is requested.